

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



日本国特許庁
JAPAN PATENT OFFICE

CZ0 15556 US/jr

29/902.172

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 6月26日

出願番号

Application Number:

特願2001-193640

出願人

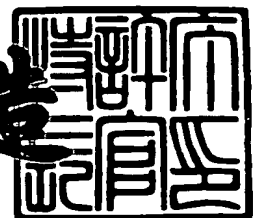
Applicant(s):

キヤノン株式会社

2001年 8月 3日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3069478

【書類名】 特許願

【整理番号】 4347015

【提出日】 平成13年 6月26日

【あて先】 特許庁長官殿

【国際特許分類】 G06K 9/36

【発明の名称】 画像処理装置、画像処理方法、プログラム並びに記憶媒体

【請求項の数】 9

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

【氏名】 戸田 ゆかり

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康徳

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100112508

【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241
【選任した代理人】
【識別番号】 100116894
【弁理士】
【氏名又は名称】 木村 秀二
【電話番号】 03-5276-3241
【手数料の表示】
【予納台帳番号】 003458
【納付金額】 21,000円
【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 0102485
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像処理装置、画像処理方法、プログラム並びに記憶媒体

【特許請求の範囲】

【請求項 1】

カラー画像データを 2 値化する 2 値化手段と、
前記カラー画像データから文字領域を検出する検出手段と、
前記文字領域中の文字を構成する N 個の色から、N 以下の M 個の色を導き出す減色手段と、
前記文字領域に対し文字きり処理を行う文字きり手段と、
前記文字きり手段により得られた文字きり単位毎に、前記 M 個の色の内、1 色を割り当てる色割当て手段と、
を有することを特徴とする画像処理装置。

【請求項 2】

前記色割当て手段は、前記文字切り単位中の文字を構成する複数画素の中で、最も多い画素数を占める色を、前記 M 個の色から選択し、前記文字切り単位中の文字を構成する全画素に割り当てることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 3】

前記減色手段は、前記カラー画像データの前記文字領域の RGB 3 次元ヒストグラムを用いて代表色を抽出する抽出手段を含むことを特徴とする請求項 1 に記載の画像処理装置。

【請求項 4】

前記減色手段は、前記抽出手段によって抽出された前記代表色のうち、1 の代表色を下地色から他の代表色への遷移的な色であるか否か判定し、遷移的な色の場合には、前記他の代表色と同色として扱うことにより、前記 M 個の色を導き出すことを特徴とする請求項 3 に記載の画像処理装置。

【請求項 5】

前記色割当て手段によって前記文字領域に割当てられた色の総数に応じて、異なる方法で圧縮する圧縮手段を更に有することを特徴とする請求項 1 乃至 4 のい

ずれかに記載の画像処理装置。

【請求項 6】

前記圧縮手段は、前記文字領域に割り当てられる色が 1 色の場合には、その文字領域に対して M M R 圧縮を行い、前記文字領域に割り当てられる色が 2 色以上、所定数 S 色未満の場合には、その文字領域に対して Z I P 圧縮を行い、前記文字領域に割り当てられる色が S 色以上の場合には、J P E G 圧縮を行うことを特徴とする請求項 5 に記載の画像処理装置。

【請求項 7】

カラー画像データを 2 値化する 2 値化工程と、

前記カラー画像データから文字領域を検出する検出工程と、

前記文字領域中の文字を構成する N 個の色から、N 以下の M 個の色を導き出す減色工程と、

前記文字領域に対し文字きり処理を行う文字きり工程と、

前記文字きり工程により得られた文字きり単位毎に、前記 M 個の色の内、1 色を割り当てる色割当て工程と、

を有することを特徴とする画像処理方法。

【請求項 8】

コンピュータに、

カラー画像データを 2 値化する 2 値化工程と、

前記カラー画像データから文字領域を検出する検出工程と、

前記文字領域中の文字を構成する N 個の色から、N 以下の M 個の色を導き出す減色工程と、

前記文字領域に対し文字きり処理を行う文字きり工程と、

前記文字きり工程により得られた文字きり単位毎に、前記 M 個の色の内、1 色を割り当てる色割当て工程と、

を実行させるためのプログラム。

【請求項 9】

コンピュータに、

カラー画像データを 2 値化する 2 値化工程と、

前記カラー画像データから文字領域を検出する検出工程と、

前記文字領域中の文字を構成するN個の色から、N以下のM個の色を導き出す減色工程と、

前記文字領域に対し文字きり処理を行う文字きり工程と、

前記文字きり工程により得られた文字きり単位毎に、前記M個の色の内、1色を割り当てる色割当て工程と、

を実行させるためのプログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は画像処理装置、画像処理方法、及びその方法を実現するプログラム並びに記憶媒体に関するものである。

【0002】

【従来の技術】

近年、スキャナの普及により文書の電子化が進んでいる。電子化された文書をフルカラービットマップ形式で記憶しようとする、例えば、A4サイズの場合では300dpiで約24Mバイトにもなり、必要なメモリが膨大になる。このような大容量のデータは、メールに添付して送信するのに適したサイズとはいえない。そこで、フルカラー画像を圧縮することが通常行われており、その圧縮方式としてJPEGが知られている。JPEGは写真などの自然画像を圧縮するには非常に効果も高く、画質も良い。しかし一方で、文字部などの高周波部分をJPEG圧縮すると、モスキートノイズと呼ばれる画像劣化が発生し、圧縮率も悪い。そこで、領域分割を行い、文字領域を抜いた下地部分のJPEG圧縮と、色情報付き文字領域部分のMMR圧縮を作成し、解凍時は白部分はJPEG画像を透過し、黒部分は代表文字色を載せて表現する方法があった。

【0003】

【発明が解決しようとする課題】

しかし、上記従来の方法では文字領域は1領域につき1色でしか表現できず、文字領域に複数の文字色が存在する場合には対応できなかった。

【0004】

本発明は上記従来技術の課題を解決するためになされたものであり、文字領域に対し複数色を割り当てる画像処理装置、画像処理方法並びにその記憶媒体を提供することを目的とする。

【0005】

【課題を解決するための手段】

上記目的を達成するため、本発明に係る装置は、
カラー画像データを2値化する2値化手段と、
前記カラー画像データから文字領域を検出する検出手段と、
前記文字領域中の文字を構成するN個の色から、N以下のM個の色を導き出す減色手段と、
前記文字領域に対し文字きり処理を行う文字きり手段と、
前記文字きり手段により得られた文字きり単位毎に、前記M個の色の内、1色を割り当てる色割当て手段と、
を有することを特徴とする。

【0006】

前記色割当て手段は、前記文字切り単位中の文字を構成する複数画素の中で、最も多い画素数を占める色を、前記M個の色から選択し、前記文字切り単位中の文字を構成する全面素に割り当てることを特徴とする。

【0007】

前記減色手段は、前記カラー画像データの前記文字領域のRGB3次元ヒストグラムを用いて代表色を抽出する抽出手段を含むことを特徴とする。

【0008】

前記減色手段は、前記抽出手段によって抽出された前記代表色のうち、1の代表色の下地色から他の代表色への遷移的な色であるか否かを判定し、遷移的な色の場合には、前記他の代表色と同色として扱うことにより、前記M個の色を導き出すことを特徴とする。

【0009】

前記色割当て手段によって前記文字領域に割り当てられた色の総数に応じて、

異なる方法で圧縮する圧縮手段を更に有することを特徴とする。

【0010】

前記圧縮手段は、前記文字領域に割り当てられる色が1色の場合には、その文字領域に対してMMR圧縮を行い、前記文字領域に割り当てられる色が2色以上、所定数S色未満の場合には、その文字領域に対してZIP圧縮を行い、前記文字領域に割り当てられる色がS色以上の場合には、JPEG圧縮を行うことを特徴とする。

【0011】

上記目的を達成するため、本発明に係る方法は、
カラー画像データを2値化する2値化工程と、
前記カラー画像データから文字領域を検出する検出工程と、
前記文字領域中の文字を構成するN個の色から、N以下のM個の色を導き出す減色工程と、
前記文字領域に対し文字きり処理を行う文字きり工程と、
前記文字きり工程により得られた文字きり単位毎に、前記M個の色の内、1色を割り当てる色割当て工程と、
を有することを特徴とする。

【0012】

上記目的を達成するため、本発明に係るプログラムは、コンピュータに、
カラー画像データを2値化する2値化工程と、
前記カラー画像データから文字領域を検出する検出工程と、
前記文字領域中の文字を構成するN個の色から、N以下のM個の色を導き出す減色工程と、
前記文字領域に対し文字きり処理を行う文字きり工程と、
前記文字きり工程により得られた文字きり単位毎に、前記M個の色の内、1色を割り当てる色割当て工程と、
を実行させる。

【0013】

上記目的を達成するため、本発明に係る記憶媒体は、

コンピュータに、
 カラー画像データを2値化する2値化工程と、
 前記カラー画像データから文字領域を検出する検出工程と、
 前記文字領域中の文字を構成するN個の色から、N以下のM個の色を導き出す
 減色工程と、
 前記文字領域に対し文字きり処理を行う文字きり工程と、
 前記文字きり工程により得られた文字きり単位毎に、前記M個の色の内、1色
 を割り当てる色割当て工程と、
 を実行させるためのプログラムを格納したことを特徴とする。

【0014】

【発明の実施の形態】

以下に、図面を参照して、この発明の好適な実施の形態を例示的に詳しく説明
 する。ただし、この実施の形態に記載されている構成要素の相対配置、表示画面
 等は、特に特定の記載がない限りは、この発明の範囲をそれらのみに限定する
 趣旨のものではない。

【0015】

<一実施の形態>

本発明に係る画像処理装置の一実施形態として、フルカラー画像を記憶媒体に
 格納する際、或は、伝送媒体上で伝送する際に、原画像の情報を残しつつ、効果
 的に圧縮する画像処理装置について説明する。

【0016】

〔概要〕

本実施形態としての画像処理装置は、まず、全画像領域について輝度ヒストグ
 ラムを生成し、2値化して、いくつかの文字領域を取り出す。次に、各文字領域
 について、文字切り処理を行い、その結果から、再度文字領域として扱うべき領
 域か否か判断する。文字領域として扱うべき領域でない場合には、その領域内の
 オブジェクトが単色かどうか判断し、単色の場合にはMMR圧縮の対象とし、単
 色でなければ、JPEG圧縮の対象とする。また、文字領域として扱うべき画像
 と判断された場合には、その領域を構成する色を所定の減色処理によって減らす

。この減色処理の結果、1色になった場合にはその色を示すパレット（例えば、 $(R, G, B) = (20, 30, 40)$ ）と、2値画像とを関連づけてMMR圧縮の対象とする。減色処理の結果、所定の色数（例えば4色）以下で表すことができる場合には、文字切り毎に、各色を示すパレットと、各色の画素位置を示す多値画像と、を関連づけてZIP圧縮の対象とする。所定の色数では表すことができない場合には、減色処理前の原画像をJPEG圧縮処理の対象とする。

【0017】

〔全体構成〕

図1に本発明を画像圧縮方法に適応した時の構成図を示す。

【0018】

101は原画像である。102は原画像を入力し、画像の最適二値化を行う画像二値化部である。103は画像二値化部により二値化された全面二値画像である。104は全面二値画像103を入力して文字領域を検出し、文字領域座標112を作成する文字領域検出部である。

【0019】

108は文字領域座標112を入力し、その座標内の原画像と二値画像を参照しながら二値画像の黒部分の原画像色を算出し複数のパレット114を作成し、それにしたがって原画像の減色処理を行う文字色抽出部である。

【0020】

105は文字領域検出部104で文字として検出された領域で、かつ文字色抽出部108にて文字色がM色未満となった領域の二値画像103の黒の領域を原画像から抜いて、その周りの色で塗りつぶし、画像Aを作成する文字部塗りつぶし部である。

【0021】

106は画像Aを入力し、縮小して画像Bを作成する縮小部である。

【0022】

107は画像Bを入力し、JPEG圧縮して圧縮コードX（113）を作成するJPEG圧縮部である。

【0023】

109は文字色抽出部108により減色された複数の文字領域の減色画像である。110は減色画像109が1ビットであるときに、減色画像を入力しMMR圧縮して複数の圧縮コードY(115)を作成するMMR圧縮部である。111は減色画像109が2ビット以上であるときに、減色画像を入力しZIP圧縮して複数の圧縮コードZ(116)を作成するZIP圧縮部である。最終的に1Aでまとめた112から116までのデータが結合してこれが圧縮データとなる。

【0024】

〔文字領域検出処理〕

図3は、文字領域検出部104での処理を説明するフローチャートである。ステップS301にてカラー画像を入力し、間引いて解像度を落しながら輝度変換を行い、輝度画像Jを作成する。例えば原画像がRGB24ビット300dpiだとすると、縦方向、横方向とも4画素ごとに

$$Y = 0.299R + 0.587G + 0.114B$$

の演算を行い、新しい画像Jを作成すると画像JはY8ビット75dpiの画像となる。ステップS302にて輝度データのヒストグラムを取り、二値化閾値Tを算出する。

【0025】

次にステップS303にて輝度画像Jを閾値Tにて二値化し、二値画像Kを作成する。更にステップS304にて黒画素の輪郭線追跡を行いすべての黒領域をラベル付けする。次に、ステップS305にて黒領域中の文字らしい領域を判定する。ステップS306にて形や位置から結合するものを結合する。

【0026】

一例を示す。例えば図4に示すカラー原稿を入力し、間引いて輝度変換したもののヒストグラムを取ると図5のようになる。このヒストグラムから平均、分散、などのデータを利用して閾値T(例えば150)を算出し、二値化した画像は図6のようになる。図6の黒画素の輪郭線追跡を行い、すべてをラベリングして、例えば、横幅が閾値以下、または高さが閾値以下の黒画素の集まりのみ文字として認識すると図7に示す黒画素の集まりが文字領域となる。ここでは説明のため図に表したが、文字領域検出処理中に、実際にこのような画像が作成されるわ

けではない。

【0027】

これらの黒画素の集まりを位置の近さや横幅、高さの一致からグループ化していくと、図8に示すような16個の文字領域が検出できる。これらの座標データが図1の文字領域座標112として格納される。

【0028】

カラー画像の2値化をおこなうのではなく、微分フィルタをかけ、すべての画素の近隣の画素とのエッジ量を算出し、そのエッジ量を二値化することにより得られた二値画像を同様に輪郭線追跡をして文字領域を検出してもよい。

【0029】

[文字領域に対する文字色抽出処理]

一方、文字色抽出部の一例のフローチャートを図19に示す。二値画像は全面二値画像103を利用したがその限りでなく、たとえば文字領域の座標とカラー画像のみ入力し、カラー画像を改めて二値化した結果を利用して代表色演算処理を行っても良い。

【0030】

この図19の処理は文字領域検出部104で文字と判定されたすべての領域に対して行われる。

【0031】

(再二値化処理)

まず、ステップS3001で再二値化の判断を行う。

【0032】

全面二値画像103は、すべての文字領域が良好に二値化されたものとは限らない。二値画像が濃すぎる場合も、薄すぎる場合も、共に結果の画質に悪影響を及ぼすので、理想的には文字領域ごとに最適二値化を行うのがよい。図5に示した全面のヒストグラムに比べて、文字領域ごとの輝度ヒストグラムは、図9のようなシンプルな形が期待できるので、閾値の決定は容易である。901は下地色の集合であり、902は文字色の集合である。本実施の形態では処理時間を節約するため、より悪影響を及ぼす「濃すぎる二値画像」に対してのみ再二値化を行

う。

【0033】

具体的には、文字領域検出部104において、文字と判定された領域の二値画像を走査し、孤立点フィルタとのパターンマッチングを行う。孤立点はその領域中に閾値以上存在するか判定し、閾値以上の場合は、その領域の輝度ヒストグラムをとり、最適閾値を算出して再二値化を行う。通常の文字領域であれば、部分的に輝度ヒストグラムを通すことによってよりよい二値画像が得られるが、まれに前よりも悪い結果（再二値化後、前より濃い二値結果になる）場合が存在する。その現象を防ぐために、再二値化では全面二値画像を得るのに使用した二値化閾値を入力し、再二値化のための閾値と比較して前よりも濃い結果が得られる場合は再二値化を行わないなどの例外処理を設ける。

【0034】

（文字切り処理）

次にステップS3002で、文字きり情報を作成する。

【0035】

文字きり部では、文字領域が横書きであるか縦書きであるかによって処理が変わる。この横書きか縦書きかの情報は文字領域検出部にて黒の固まりの並びから判断され作成される。横書きである場合は、まず主走査方向に二値画像の黒画素の射影を取る。そして行の切れ目を検出した後、行ごとに副走査方向に黒画素の射影を取り、1文字毎の情報を得る。縦書きである場合は行きりだしが副走査方向に行われ文字きりだしが主走査方向に行われる。この時、多少の画像の傾きにも耐えられるように、行きりだしの射影を取るのに行方向に3分割して行うなどとよい。この処理によって、各行の座標情報、および各行に存在する文字の座標情報を得ることが出来る。

【0036】

一方、文字判断処理（後述のステップS3003）では、文字領域検出部にて文字と判定された領域内の黒オブジェクトそれぞれを、さらに文字であるかどうかを判定するためにも文字きり情報を利用する。具体的には1文字の大きさや形から文字であるか否かを判定する。画質、圧縮という点からかんがみると、単色

、または複数色に変換するのに「文字である」ことにこだわる必要はないのだが、（たとえば、単色で表現されたマークなどはJ P E Gで表現するより単色M M Rで表現したほうが画質圧縮率ともによいのだが）、確率の問題として、文字以外の領域はグラデーションで表現されていることが多いからである。

【0037】

（文字判断処理）

次に、ステップS3003で文字判断を行う。

【0038】

ここでは、文字きり（S3002）での情報を入力し、行ごとにその行の平均文字サイズを演算する。このとき極端に小さい文字の情報は無視するとよりよい結果が出る。その平均サイズよりも極端に大きい文字矩形は文字でない、と判定する。また、平均に関わらず、縦横比情報などから明らかに文字でない形状の場合は文字でない、と判定する。

【0039】

その領域中にm文字存在する場合、m文字すべてが文字でないという判定になった場合、文字判定部はこの領域は画像である、という結果を出力する。

【0040】

しかしながら、m文字中n文字（ $m > n$, $n \geq 0$ ）が文字でない、すなわち文字である矩形が残っている場合は、二値画像上から文字でないと判定された黒オブジェクトを消去して、この領域は文字である、という結果を出力する。

【0041】

念の為最終画質をかんがみて、次の例外処理を加える。たとえば領域中に10文字存在するときに飛び飛びの5文字が文字として単色化され、残りが文字でないとしてJ P E G圧縮されると、「ムラ」な画像が得られ、目視的に好ましくない画像となる。そこで、この文字判断部にて、文字、画像の判断が頻繁に切り替わるようなケースには、その並びや文字と判断される矩形の頻度などから全矩形文字にそろえる。または全矩形画像にそろえる。

【0042】

文字判断処理で、文字と判断された領域についてはステップS3004に進み

、文字でないと判断された領域については、ステップS3005に進む。

【0043】

(単色判断)

ステップS3005では単色判断を行う。

【0044】

この処理を通る領域は文字領域検出部にて文字と判定されたものの、文字判断にて文字でないと判定された領域である。前述したように、文字であるか否かにかかわらず、単色で表現されている領域であれば単色化してMMRしたほうが、画質圧縮率ともによい。そこでこの領域は単色であるか否かの判断を行う。

【0045】

具体例としては二値画像の黒部に値するカラー画像の画素のRGBそれぞれのレベルのヒストグラムを取りそれぞれのヒストグラムすべての分散値が閾値以下であればこの領域は単色であると判定する。

【0046】

単色である場合にはステップS3006の1色抽出にすすみ、複数色である場合にはendにすすむ。

【0047】

(一色抽出処理)

ステップS3006での1色抽出処理を図12のフローチャートを用いて説明する。

【0048】

ステップS1201にてその文字座標が参照する二値画像の細線化処理を行い、スキャナ読み込み時の下地から文字部への変化部にあたる黒を減らしていき、新しい二値画像newbiを作成する。次にステップS1202にてnewbiの黒画素に対応する原画像のRGBのそれぞれの値のヒストグラムを取る（もちろん、YUV等の他の色空間でも構わない）。ステップS1203にてRGBそれぞれの代表値を算出する。例えば、一番大きな値でも良い。またはヒストグラムのステップ数を少なくして大まかなヒストグラムで一番大きな値を得た後に、このヒストグラムに存在する細かなヒストグラムで一番大きな値を得る方法でも

よい。

【0049】

後者のような方法を取ることににより、図13に示したようなヒストグラムからノイズ1302に惑わされることなく真の代表値1301を得ることができる。図を使用して詳細に説明をすると、細かなヒストグラムとしては例えば8ビットのRデータから256段階のヒストグラム（図13に示す）が得られるわけだが、これの最大値は1302になり、これは真の代表値ではない。そこで、ヒストグラムをオーバーラップする64の幅にわけ、8段階のものを256段階のヒストグラムから再計算する。それを0から8に示したが、0と8は32幅しかない。この再計算により代表値は⑥に存在することがわかり、⑥内の最大値を検索して1301を得ることができる。以上の処理をすべての文字座標に繰り返すことにより、すべての文字座標に1つずつ代表色が算出される。

【0050】

（減色処理）

ステップS3004では、文字に対し、減色処理を行う。

【0051】

減色処理部1082においては、文字部の色について、元原稿が単色で表現されていた場合においても、スキャナ読み取り時に下地から文字部への色の遷移部が存在する。

【0052】

図22、図25にスキャナ読み取りによる色の遷移の様子を示す。図22では簡単のため、RGBで説明せず、Rのみで説明する。Aという文字はもともとR=32レベルの単色にて構成されていたものであるが、スキャナにより読み込むと、そのデータは拡大の画素で示すようにばらついてしまう。もともとのレベルR=32近くの黒まで達している画素は3201, 3202, 3203の3つだけに過ぎず、他の画素は、下地色（この場合白）とR=32の間のレベルにとどまる結果、遷移部であるグラデーションで文字が表現されている。図25では図22の文字Aの画素レベルの3次元ヒストグラムにて色の遷移の様子を示す。下地色が3501に示す白であり、文字色が3502に示す黒であるとする。

遷移部を3503に示す。

【0053】

元々単色で表現されていた文字部のスキナ読み込みによるバラツキである遷移部を厳密に表現する必要はない。その代表色のみで表現することが出来れば画質もよく、データ量も少なくてすむ。しかしながら、二値画像の細線化などを行ったとしても、この下地から文字部の遷移部の色を完全に除去することは難しい。そこで、1文字は単色であらわされることが多いことを利用して、文字きり情報を使い、1文字1色に限定することにより画質、圧縮率の向上を目指す。ただし、もともとからグラデーションで表現される文字などを更に高画質で圧縮したい場合には、その文字が複数色で表現されているかの判定など例外処理を加えればよい。即ち、このように文字切り情報を利用して1文字1色にすれば、元々単色で表現されていた文字画像のスキナ読み込みによるバラツキとして発生する遷移部を除去する事ができる。

【0054】

次に減色処理部の詳細説明を図20のフローチャートを利用して行う。

【0055】

ステップS3101にてその文字座標が参照する二値画像の細線化処理を行い、スキナ読み込み時の下地から文字部への遷移部にあたる黒を減らしていき、新しい画像thinimageを作成する。thinimageは二値画像であるが、後述のステップS3110の処理にて使用するので255（黒）と0（白）の二値として8ビットで構成する。次にステップS3102にてthinimageの黒画素に対応する原画像のRGBの3次元ヒストグラムを取る。この際、普通にヒストグラムをとると、たとえば入力画像がRGB各8ビットだとすると、 $256 \times 256 \times 256$ のヒストグラムが必要になる。文字部に必要なのは解像度であり、階調は必要ないこと、また、スキナによる読み込み時のばらつきを押さえながら代表色を算出するには多少の画素値の違いは無視した方がよいこと、などをかんがみると、これほどの細かなヒストグラムは必要ない。たとえば、したがってこの例では、上位5ビットのRGB 3次元ヒストグラムをとる。このヒストグラムをとる際は、その文字領域に存在する黒画素の総数black

numも算出する。

【0056】

本実施の形態ではRGB空間を利用したが、LabやYUVなどの他の色空間でも構わない。また、3次元ヒストグラムをとったが、各色それぞれの1次元ヒストグラムを3つとってもかまわない。

【0057】

ステップS3103にてその領域に存在する文字色数を示すcolnumを0にリセットする、処理済の画素数を示すokpixelを0にリセットするなどの初期化作業を行う。

【0058】

ステップS3104にて代表値を算出する。ここでは、注目ヒストグラムを中心とする7つのヒストグラムの合計値が最大な点を代表値とする。(7つのヒストグラム：注目点と、R次元で隣り合った2つ、G次元で隣り合った2つ、B次元で隣り合った2つ(図15に示す))このように検出された最大値をcolR[colnum]、colG[colnum]、colB[colnum]に代入する。

【0059】

次にステップS3105にてこの代表値を中心としたこの代表値に変換されるべき色の範囲を決定する。

【0060】

代表値をそれぞれ固定して3つの1次元ヒストグラムを得る。図23にその様子を示す。例えば、代表値を(Col R(26), Col G(30), Col B(22))と仮定すると3次元ヒストグラムのGを30、Bを22に固定したRの1次元ヒストグラム(=全ヒストグラムを3301の線に射影)、3次元ヒストグラムのRを26、Bを22に固定したGの1次元ヒストグラム(=全ヒストグラムを3302の線に射影)、3次元ヒストグラムのRを26、Gを30に固定したBの1次元ヒストグラム(=全ヒストグラムを3303の線に射影)を得る。例えばRの1次元ヒストグラムは図24のような形になるがこれの3401の点、3402の点を検出しそれを代表値に題する「Rの範囲」とする。検

出の方法としては、画像の二値化閾値決定方法を利用する方法などがある。例えば3403は代表値であるが、0から代表値までのヒストグラムを二値化閾値決定関数に代入する事により3401の点を得て、代表値から31までのヒストグラムを反転したものを二値化閾値決定関数に代入する事により3402の点を得る事が出来る。

【0061】

この色の範囲をR, G, Bそれぞれ決定し `fg_range[colnum]` に代入する。

【0062】

ステップS3106にて `fg_range[colnum]` 内の3次元ヒストグラムの値をすべて0にする。この時、0にした画素の数を処理済の画素数を示す `okpixel` に加算する。

【0063】

ステップS3107にて近似色判定を行う。近似色判定は、`fg_color[colnum]` に対して、それまでに出現した色すべて(`fg_color[0]` から `fg_color[colnum-1]` まで) に対して行う。ただし、近似色が見つかった場合はループを抜ける。文字きりの説明にて言及したようにスキャナにより読み取った画像は下地色と文字色の間に遷移的な色が発生してしまう。図25に示すように下地色(3501)と文字色(3502)の間の色の画素(3503)が存在してしまうのである。二値画像を参照する事により、3504にあらわした線より下地色側の色は3次元ヒストグラムに加算されない。

(また二値画像の細線化により3504の線はより文字色側による) しかしながらなお遷移部分は存在していて、3502の文字色抽出後に遷移部分の色を抽出してしまうことがある。例えば、`fg_color[0]` が(32, 40, 40)なのに対して、`fg_color[m]` に(96, 112, 96)を取り出してしまう事がよくある。これを同じ文字色だと判断するために近似色判定を行う。均等色空間でないRGB空間では近似色判定がやや難しいので、ここでは、Lab空間にて判定する。`fg_color[0] = (32, 40, 40)` をLab変換すると(15, -4, -1)であり、`fg_color[`

m] = 96, 112, 96) の Lab 変換は (45, -9, 7) である。

【0064】

これにより、元々単色で表現されていた文字画像のスキナ読み込みによるバラツキとして発生する遷移部を除去する事ができ、高画質高圧縮の画像圧縮を可能とする。

【0065】

本来下地色も検出し、下地色と fg_color[0] の延長線に存在する色を近似色とするべきであるが、今回は下地は白が多いと割り、Lab の ab の距離が閾値以下であったら近似色であると判定する。判定結果は、kinji[] 行列に残す。kinji[] にて同じ数字を持つものは近似色であることを表す。

【0066】

例を以下にしめす。kinji[0] と kinji[3] はともに 0 である。すなわち、fg_color[0] と fg_color[3] は近似色であるということである。

【0067】

```
fg_color[0] = (32, 40, 40)      kinji
[0] = 0
fg_color[1] = (248, 64, 48)      kinji
[1] = 1
fg_color[2] = (48, 256, 32)      kinji
[2] = 2
fg_color[3] = (96, 112, 96)      kinji
[3] = 0
```

ステップ S3108 で colnum をインクリメントする。

【0068】

ステップ S3109 にて色抽出済みの黒画素が 75% 以上超えたかどうか判定する (75 は一例)。

【0069】

すなわち $okpixel \times 75 > blacknum \times 100$ を満たすかどうかチェックする。

【0070】

超えていたらステップS3110にすすみ超えていないときはステップS3113にすすむ。

【0071】

ステップS3110では *thinimage* 上にパレット画像を構築する。具体的には *thinimage* の値が255の画素（まだ色割り当てが行われていないことを示す）に対応するカラー画像の画素RGBレベルを参照し、そのRGBデータが *fg_range* [m] 内に存在する場合は *kinji* [m] + 1 の値（すなわち1）を *thinimage* の対応する画素値に当てはめる。ここで、*kinji* [m] でなく *kinji* [m] + 1 を代入するのは、0が文字部でない（下地）ことをあらわす特別な数字であるため *kinji* [m] が0の時そのまま代入できないからである。

【0072】

ステップS3110にて *thinimage* 上にパレット画像を作成すると、次にステップS3111にて文字きり情報を利用して、パレット画像から文字毎の色情報 *charpal* を作成する。

【0073】

ここで、文字切り単位毎の色情報 *charpal* の作成方法について、図26の文字画像を例にして説明する。この文字の「明日は晴れです。昨日は雨でした。」の「晴れ」は *fg_color* [1] で抽出された赤色、「雨」が *fg_color* [2] で検出された青色であり、その他は黒であるとする。

【0074】

文字きり処理により、1文字目の「明」が最初の1文字として処理される。この文字に存在する細線画像 *thinimage* の黒画素数は100個とする。そのうち70個が *fg_range* [0] 内に存在し、20個が *fg_range* [3] 内に存在したとする。 *kinji* [0] も *kinji* [3] もともに0（

近似色である)であるので、90画素ともにステップS3110にて1が割り当てられる。ここでは、存在する黒画素の最大数を1つの数字が占めたらその色を割り当てることとする。このケースでは、まだ色が割り当てられない画素は残り10画素であり、この10画素が全てこれから検出される色だったとしても90を超えることはない。そこで、「明」の色情報`charpal[0]`は1とする。

【0075】

このように各文字切り部分について、その文字に割り当てる色を0~2の3つの`charpal`から選択する。その結果、「晴れ」の左辺「日」右辺「青」「れ」は赤である`kinji[1]+1=2`、「雨」は青である`kinji[2]+1=3`が割り当てられ、残りの文字は黒である`kinji[0]+1=1`が`charpal`として割り当てられる。

【0076】

この例では、存在する文字数20文字のすべてに対応する`charpal`に色情報が割り当てられたので、ステップS3113の条件にて文字色抽出ループを抜ける条件に当てはまり、ステップS3114に飛ぶ。しかし、ここでたとえば20文字中18文字しか当てはまらなかった場合、ステップS3112にて`charpal`が作成されない文字のみの3次元ヒストグラムを取り直してステップS3105に戻る。この際`blacknum`も`charpal`未作成の文字の細線黒画素数が代入され、`okpixel`は0にリセットされる。そして上記の例でいうと、`fg_color[3]`まで検出されているので`fg_color[4]`から同様に抽出を行う。

【0077】

この時、高速化のため3次元ヒストグラム取り直しステップS3112は一度限りとし、ステップS3110、ステップS3111は2度までと限定してもよい。実験によるとそれで十分な画質が得られている。

【0078】

上記の限定を入れた場合、ステップS3113にてループを抜けても`charpal`が未作成の文字がありえる。そこで、ステップS3114において、`ch`

arpal未作成の文字数nokoricharが1以上の時、強制的に未作成charpalに色情報を割り当てる。具体的には、ステップS3111の割り当てではthinimageの画素値が255の画素（すなわちまだ色抽出が行われていない画素）のその後の動向も考えて、最大値が得られた場合のみにcharpalに色情報を代入した。しかしステップS3114においてはthinimageの255の画素は無視して残りの0（下地）以外の画素の最大値でcharpalを決定する。ただし、0（下地）以外の画素がすべて255のケースもある（1画素も色抽出が行われなかった）。その場合には、位置的に近い文字のcharpalの色を代入する。たとえば、図26の「す」がそのケースの場合に、位置的に近い「で」のcharpalを代入する。

【0079】

ここまでで抽出した色数としてcolnumという数値が選られている。しかし、これは近似色も含めているし、抽出したものの使用しない色もありえるので、実際に使用されている色数（usecolnum）とは異なる。そこで、ステップS3115にてcharpalをなめて実際に使用されている色数を計算する。

【0080】

ステップS3116ではステップS3115にて計算されたusecolnumが16以上の時は下地（0）とあわせて17色存在することになり、4ビットであらわせないで、ステップS3117に進み、この領域の減色処理はあきらめてDOJPEGを返す（下地画像として表現する）。ただし、ここで4ビットでなく8ビットまで許した場合は16でなく256になる。

【0081】

usecolnumが1の場合は、ステップS3118に進み、使用色1色のパレットを作成し、ステップS3119にて入力二値画像の切抜きをおこない、ステップS3120にてDOMMRを返す。

【0082】

usecolnumが2以上16未満の場合は、ステップS3121に進み、使用色のパレットを作成し、ステップS3122にてパレット画像を作成する。

ここでは、2, 3のケースでは、1画素2ビットにて表されるパレットを、4以上16未満のケースでは1画素4ビットにて表されるパレットを作成する。そしてステップS3123にてDOZIPを返す。

【0083】

このようにして作成された減色画像109が1ビットの場合（文字色抽出によりDOMMRが返された場合）は110にてMMR圧縮し、圧縮コードYを作成する。また、減色画像109が2ビット以上の場合（文字色抽出によりDOZIPが返された場合）は111にてZIP圧縮し、圧縮コードZを作成する。DOJPEGが返された場合は減色画像が存在しないMMR圧縮部110やZIP圧縮部111には送らず、文字部塗りつぶし部105に文字領域として扱わないようにコマンドを送る。

【0084】

〔文字塗りつぶし処理〕

文字部塗りつぶし部104の処理の一例を図10と図11を用いて説明する。図11は、文字部塗りつぶし処理の流れを示すフローチャートである。

【0085】

一例として、グラデーション画像を背景とし、ABCという青色の文字が中央付近に描かれた、図10(a)のような画像を原画像とする。この原画像から(b)のような1つの文字領域の2値画像を得たとする。そして、文字部塗りつぶし処理として、まず、ステップS1101で、全画像を 32×32 の領域（以下、パーツ）に分割し、パーツごとに処理をおこなう。図10(c)にパーツごとに分けた様子を示す。この図では簡単に説明するため、 5×5 のパーツに分割した状態を示している。各領域の左上の数字はパーツ番号を示す。この領域分割の個数はこれに限るものではなく、他の数でもよい。

【0086】

ステップS1102では未処理のパーツか否か判断し、未処理のパーツについては、ステップS1103に進んで文字塗りつぶし対象領域がそのパーツ内に存在するか否か判断される。文字領域検出部104で文字領域と判定された領域でも、文字色抽出部108でDOJPEGが返された領域は、文字部塗りつぶしの

対象領域とはしない。

【0087】

図10(c)の例では、パーツ00~04, 10, 14, 20, 24, 30~35は、ステップS1103において文字塗りつぶし対象領域がないと判断され、処理は行われずに、次のパーツに進む。文字塗りつぶし対象領域の存在するパーツ（例えばパーツ11）に対しては、ステップS1104に進み、対応する二値画像を参照し、二値画像の白部分に対応するカラー画像のRGB値（またはYUV等でも良い）の平均値ave_colorを算出する。次にステップS1105において、対応する二値画像を参照し、黒画素に対応する画素の濃度データをave_colorとする。以上の処理を文字塗りつぶし対象領域の存在するパーツ（ここでは、パーツ12, 13, 21, 22, 23）に繰り返す。このようにして、文字の存在した部分に周りの画素の平均値を埋めることができる。

【0088】

この画像を縮小部106にて縮小する。本実施の形態では単純間引きとする。ちなみに、この縮小と文字部塗りつぶし処理は順番を逆にしても構わない。その場合二値画像とカラー画像の位置のずれを気を付ける必要がある。

【0089】

文字領域座標112、パレット114、圧縮コードX(113)、圧縮コードY(115)、圧縮コードZ(116)の5つをまとめたフォーマットを必要ならば作成する。5つをまとめるフォーマットの一例としてAdobe(商標)のPDFなどが考えられる。AdobeのPDFとはAdobeが無償配布しているAcrobat Reader(商標)というアプリケーションで表示できるフォーマットであり、ドキュメントを作成したアプリケーションがないために、受け手側でファイルを開けないなどのトラブルを避けることが出来る。その他のフォーマットとしては、XMLなどがある。XMLとはネットワークを介して文書やデータを交換したり配布したりするための記述言語である。

【0090】

〔伸長処理〕

図2に伸長処理を行うために必要な構成を示す図である。

【0091】

201は圧縮コードX(113)を入力し、JPEG伸長処理をおこない多値画像Eを作成するJPEG伸長部である。202は多値画像Eを入力し、拡大処理を行う拡大部である。203は拡大部202により拡大された多値画像Fである。204は圧縮コードY(115)を入力し二値画像G(205)を作成するMMR伸長部である。206は圧縮コードZ(116)を入力し多色画像H(207)を作成するIP伸長部である。208は文字領域座標112とそれに対応するパレット114および二値画像G(205)または多色画像H(207)を入力し、二値画像または多色画像の画素データが透過をあらわす場合は画像F(203)の画素の色を、それ以外の場合は対応するパレット色を選択し最終的な画像である画像I(209)を作成する画像合体部である。

【0092】

図14に合体処理208の結果例を示す。まず図14(a)に圧縮コードCのJPEG伸長結果を示す。これは、図10の画像を利用したが、JPEG圧縮の量子化非可逆方式を利用すると図10(c)とは微妙に画素値が異なるデータとなっている。しかし、文字部を抜く前の原画像をJPEG非可逆圧縮方式で圧縮する場合と比較して、同じ量子化テーブルを利用した場合において画素値の変化は少ない。すなわち高画質である。本例では合体処理を行う文字領域画像はMMR圧縮された二値画像とする。伸長された二値画像を(b)に示す。そのパレットはR=20 G=30, B=225とする。二値画像(b)を参照して黒画素の対応するところの画像(a)上にパレット色(20, 30, 255)データをのせ、最終的に(c)のような画像が出来上がる。多色画像の場合はパレット数が変わり、たとえば2ビットなら00, 01, 10, 11の4つの画素値に割り当てられたパレットを当てはめていく。そのうち1つは透過を示し、たとえば00とすると、00の値をもつ画素は画像(a)の画素を選択する。

【0093】

01の時は01のパレット値、10の時は10のパレット値、11のときは11のパレット値をのせる。このようにして伸長画像209が作成される。

【0094】

＜他の実施形態＞

上記実施の形態では二値画像は全面単一閾値にて作成したがその限りでなく、たとえば、文字領域検出104により検出された文字領域ごとに最適閾値を演算して二値画像を作成してもよい。その場合、図19のステップS3001の再二値化判断が必要なくなる。

【0095】

また、文字部塗りつぶし、105、文字色抽出108にて同じ二値画像を使用した、その限りでなくそれぞれに最適な二値化部を105、108内部にて所有してもかまわない。

【0096】

また、上記実施形態では、輝度の低い下地上の輝度の高い文字（反転文字）の処理が不可能であるが、たとえば図16に示す構成にすれば可能となる。1702は微分処理部であり、図17に示すような微分フィルタを注目画素を中心にか、その絶対値が閾値を超えたら黒、超えなかったが白というように二値化していく。図17（a）は1次微分フィルタであり、上は横線を検出することができ、下は縦線を検出することができる。2つのフィルタの絶対値の合計を利用すると斜め線を検出することができる。また、斜め線フィルタを利用してもよい、図17（b）は二次微分フィルタで全方向に対応したものである。二次微分フィルタも横方向、縦方向、と作成することも可能である。このようなフィルタを全画素にか、微分画像1702を作成する。この時、全画素でなく、間引きながらフィルタをかけることによって同時に解像度も落とすことも可能である。以上のように作成された二値画像に図3の304からの処理を行えば反転文字も含んだ文字領域座標も検出することができる。また、反転文字も対象にした場合は二値化部1703も対応しなければならない。反転文字領域も文字領域として抽出された場合、上記実施の形態では図9のパターンしか入ってこないと想定していたが、図18の3パターンが主に入ってくることになる。（b）が反転文字であり、（c）が同一のグレー下地上に黒文字と白文字の2色が存在するケースである。これらの3パターンを考えて、二値化部1703では、A点とB点を検出し、AとBにはさまれた領域は白、その他は黒の二値化処理をおこなうとよい。または

、（c）のケースはかんがえずに下地と文字部を分ける1つの閾値を検出し、反転パターンであれば反転する処理を行えばよい。このように反転文字領域も対応すればJ P E G圧縮される画像上には実施の形態1では残ってしまっていた反転文字領域も文字部塗りつぶしによりスムージングされるので、圧縮効率もよく、またその反転文字部も解像度やモスキートノイズの劣化なしに圧縮することが可能となる。

【0097】

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0098】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはC P UやM P U）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（O S）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0099】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるC P Uなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは

言うまでもない。

【 0 1 0 0 】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した（図 1 9 および／または図 2 0、2 1 に示す）フローチャートに対応するプログラムコードが格納されることになる。

【 0 1 0 1 】

【発明の効果】

本発明によれば、文字切り単位毎に 1 色の色を割り当てるため、文字を効率的に複数色で表すことができ、圧縮システムに利用すると、高画質高圧縮の画像圧縮を可能とする。

【図面の簡単な説明】

【図 1】

図 1 は本発明の一実施形態に係る画像処理装置の構成を示す図である。

【図 2】

図 2 は本発明の一実施形態に係る画像処理装置で圧縮したデータを伸長するための構成を示す図である。

【図 3】

図 3 は本発明の一実施形態に係る文字領域検出処理の一例のフローチャートである。

【図 4】

図 4 は本発明の一実施形態に係る文字領域検出処理を説明する図である。

【図 5】

図 5 は本発明の一実施形態に係る文字領域検出処理を説明する図である。

【図 6】

図 6 は本発明の一実施形態に係る文字領域検出処理を説明する図である。

【図 7】

図 7 は本発明の一実施形態に係る文字領域検出処理を説明する図である。

【図 8】

図 8 は本発明の一実施形態に係る文字領域検出処理を説明する図である。

【図 9】

図 9 は本発明の一実施形態に係る文字領域の再 2 値化処理を説明する図である。

【図 1 0】

図 1 0 は本発明の一実施形態に係る文字塗りつぶし処理を説明する図である。

【図 1 1】

図 1 1 は本発明の一実施形態に係る文字塗りつぶし処理を説明するフローチャートである。

【図 1 2】

図 1 2 は本発明の一実施形態に係る 1 色抽出処理を説明するフローチャートである。

【図 1 3】

図 1 3 は本発明の一実施形態に係る 1 色抽出処理を説明する図である。

【図 1 4】

図 1 4 は本発明の一実施の形態に係る画像処理装置で圧縮したデータを伸長し、合体する様子を説明する図である。

【図 1 5】

図 1 5 は本発明の一実施の形態に係る減色処理を説明する図である。

【図 1 6】

図 1 6 は本発明の他の実施形態に係る画像処理装置の構成を示す図である。

【図 1 7】

図 1 7 は本発明の他の実施形態に係る画像圧縮処理を説明する図である。

【図 1 8】

図 1 8 は本発明の他の実施形態に係る文字領域の 2 値化処理を説明する図である。

【図 1 9】

図 1 9 は本発明の一実施の形態に係る文字色抽出処理を示すフローチャートである。

【図 2 0】

図 2 0 は本発明の一実施の形態に係る減色処理を示すフローチャートである。

【図 2 1】

図 2 1 は本発明の一実施の形態に係る減色処理を示すフローチャートである。

【図 2 2】

図 2 2 はスキャナにより発生する文字部の遷移部（グラデーション）を説明する図である。

【図 2 3】

図 2 3 は本発明の一実施の形態に係る減色処理を説明するための図である。

【図 2 4】

図 2 4 は本発明の一実施の形態に係る減色処理を説明するための図である。

【図 2 5】

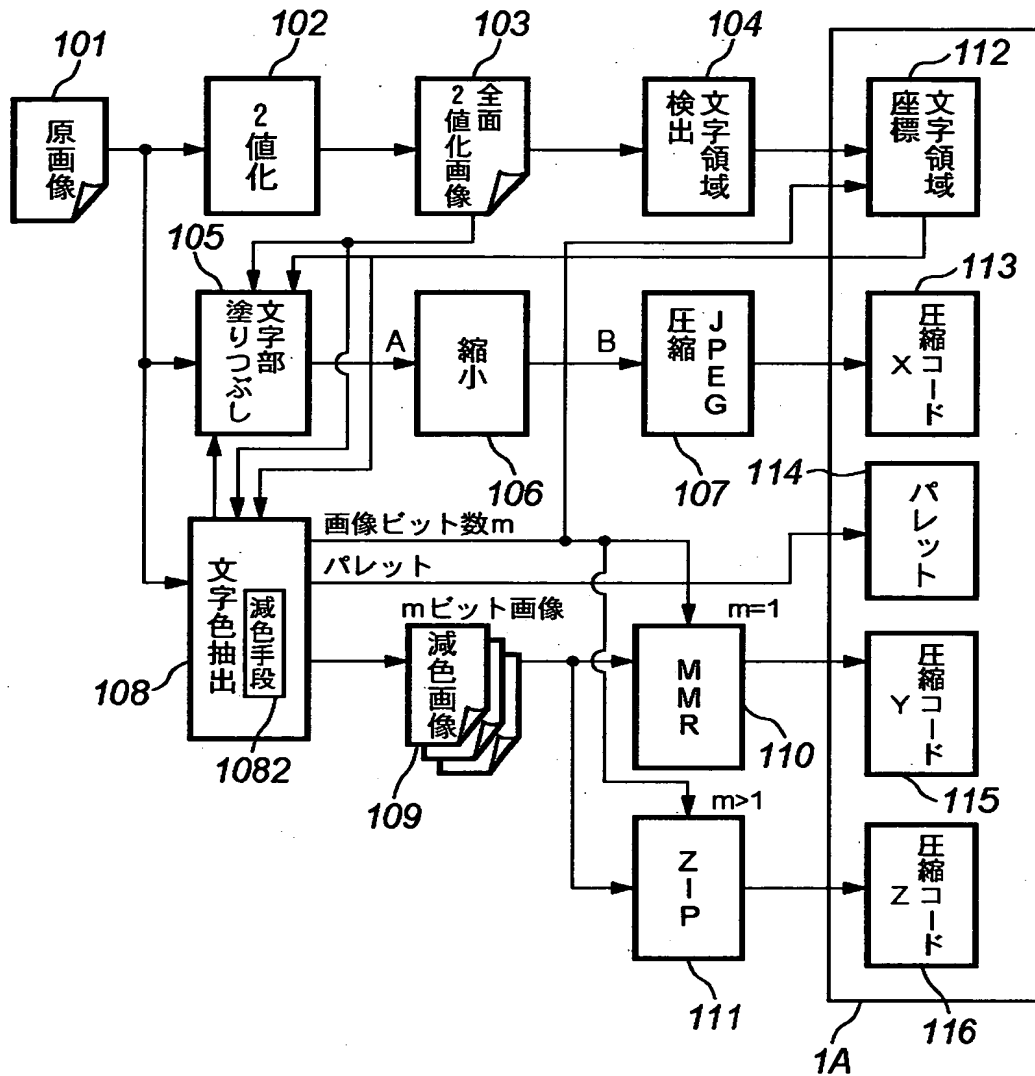
図 2 5 はスキャナにより発生する文字部の遷移部（グラデーション）を 3 次元ヒストグラムで表した図である。

【図 2 6】

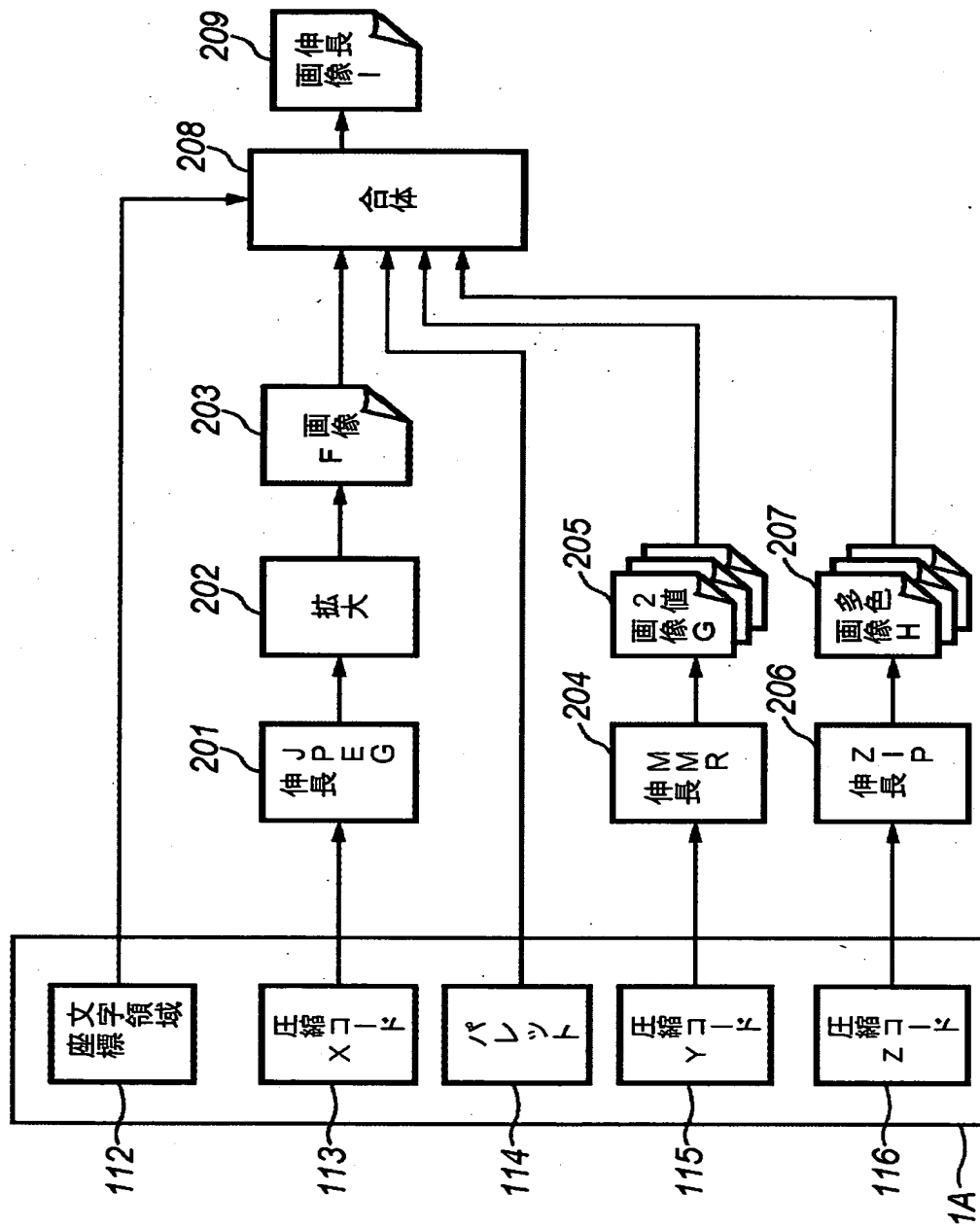
図 2 6 は本発明の一実施の形態に係る、文字きり情報を利用して文字の色を決定する色割当て処理を説明する図である。

【書類名】 図面

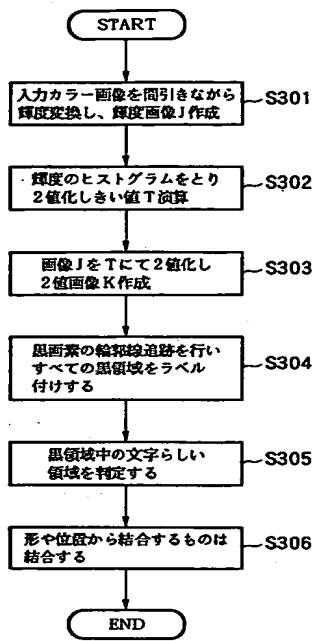
【図 1】



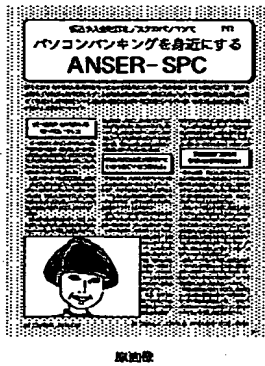
【図2】



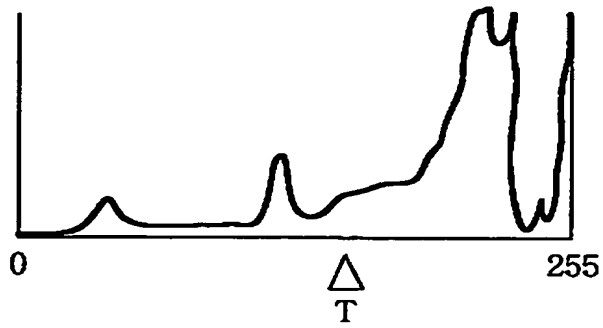
【図 3】



【図4】



【図 5】

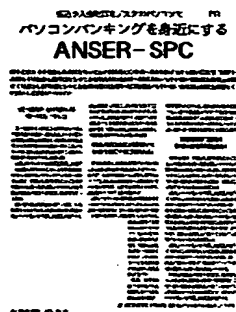


輝度ヒストグラム

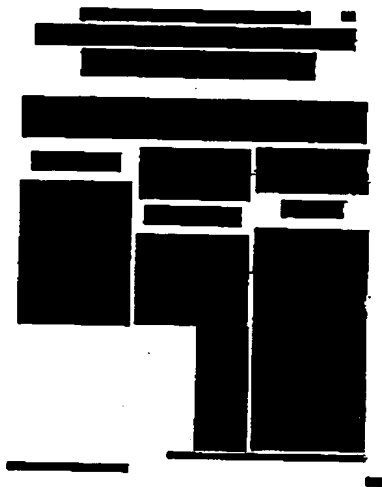
【図 6】



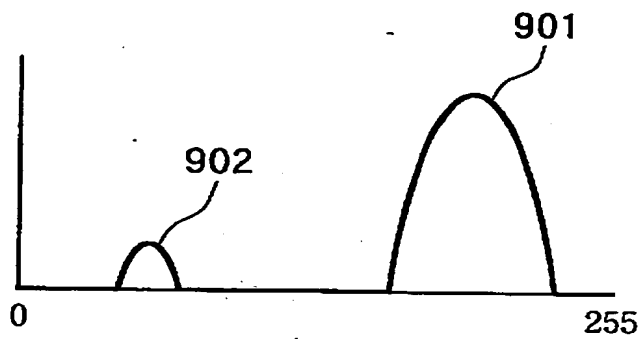
【図 7】



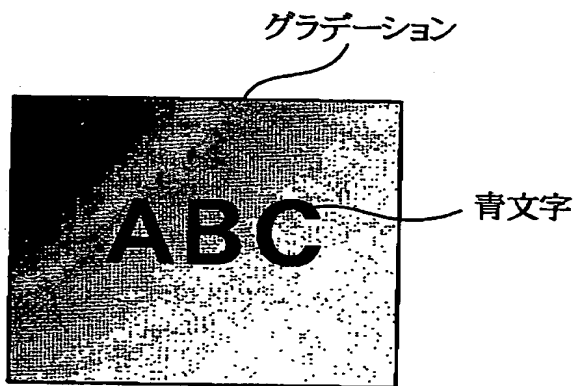
【図 8】



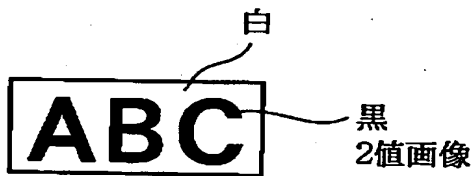
【図 9】



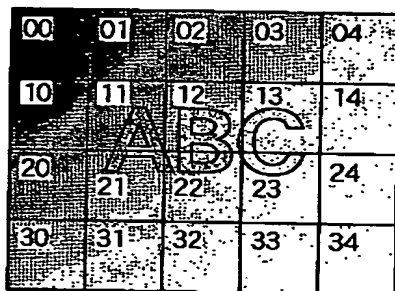
【図 10】



(a)

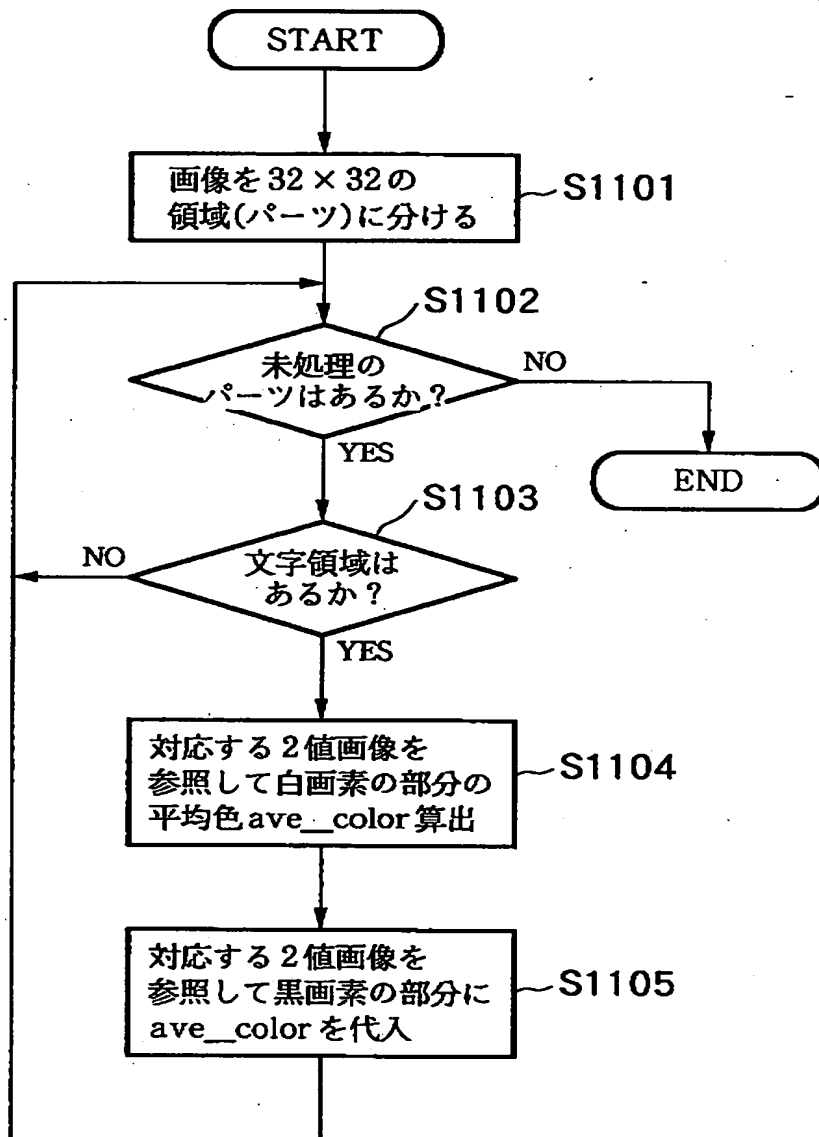


(b)

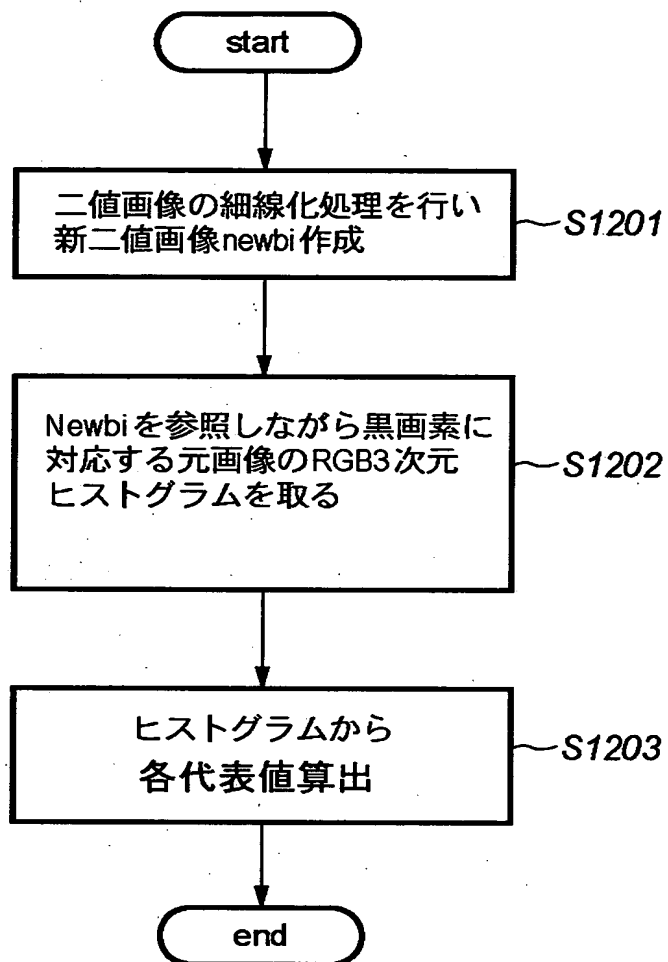


(c)

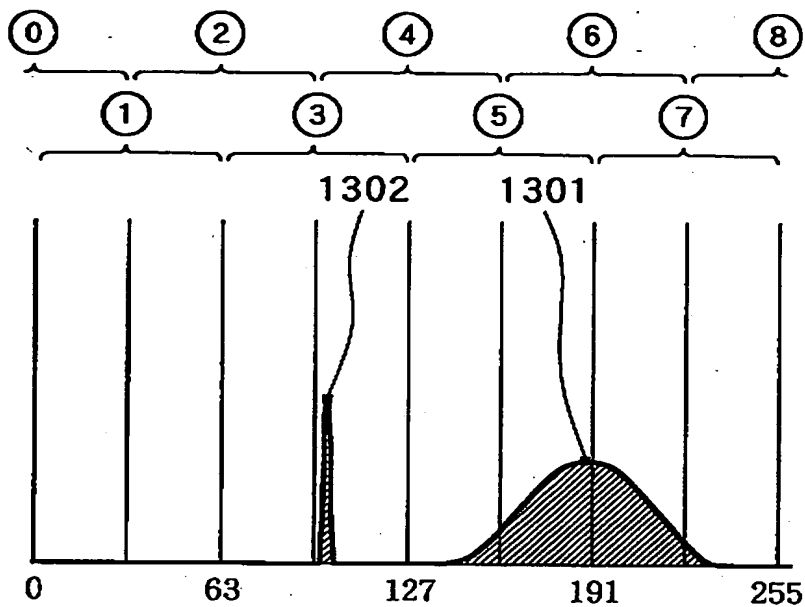
【図 11】



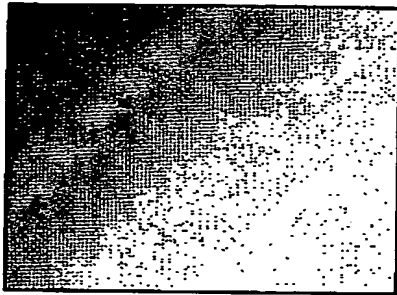
【図 12】



【図 13】



【図 14】

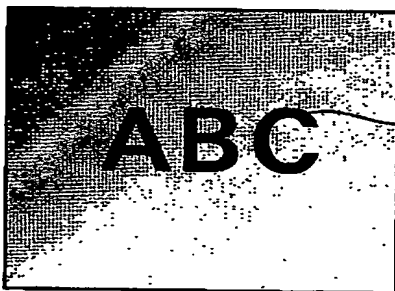


(a)

ABC

代表色データ
R=20,G=30,B=225

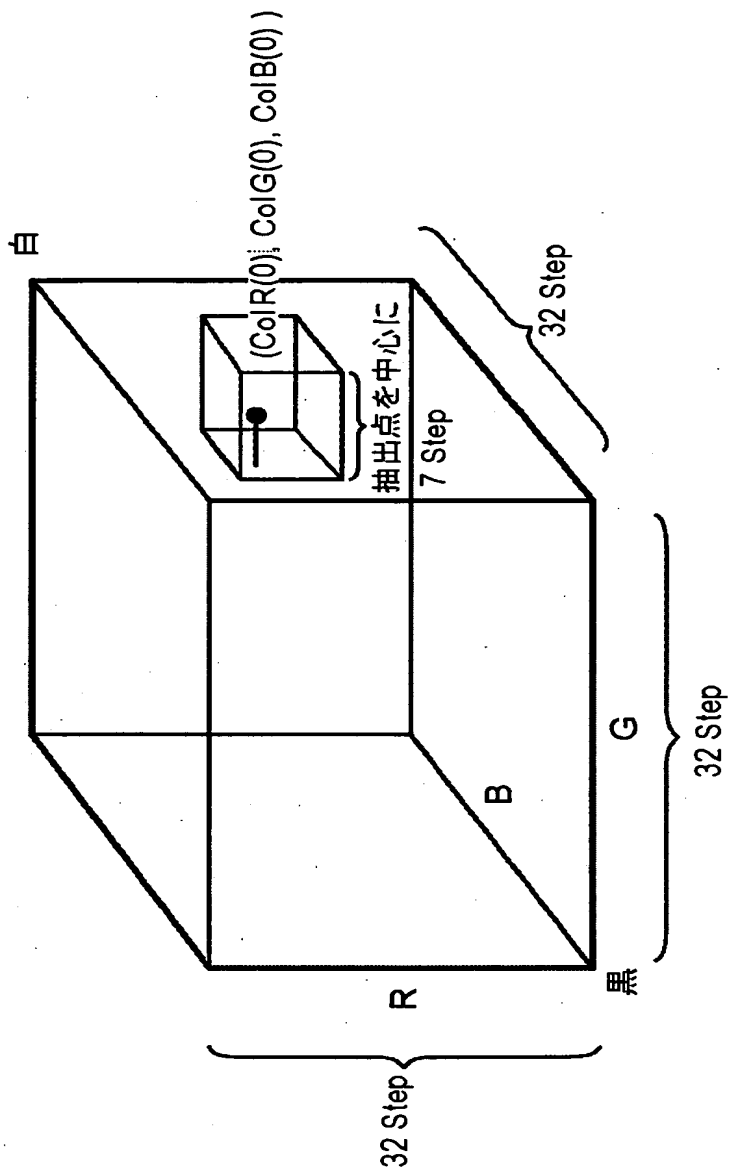
(b)



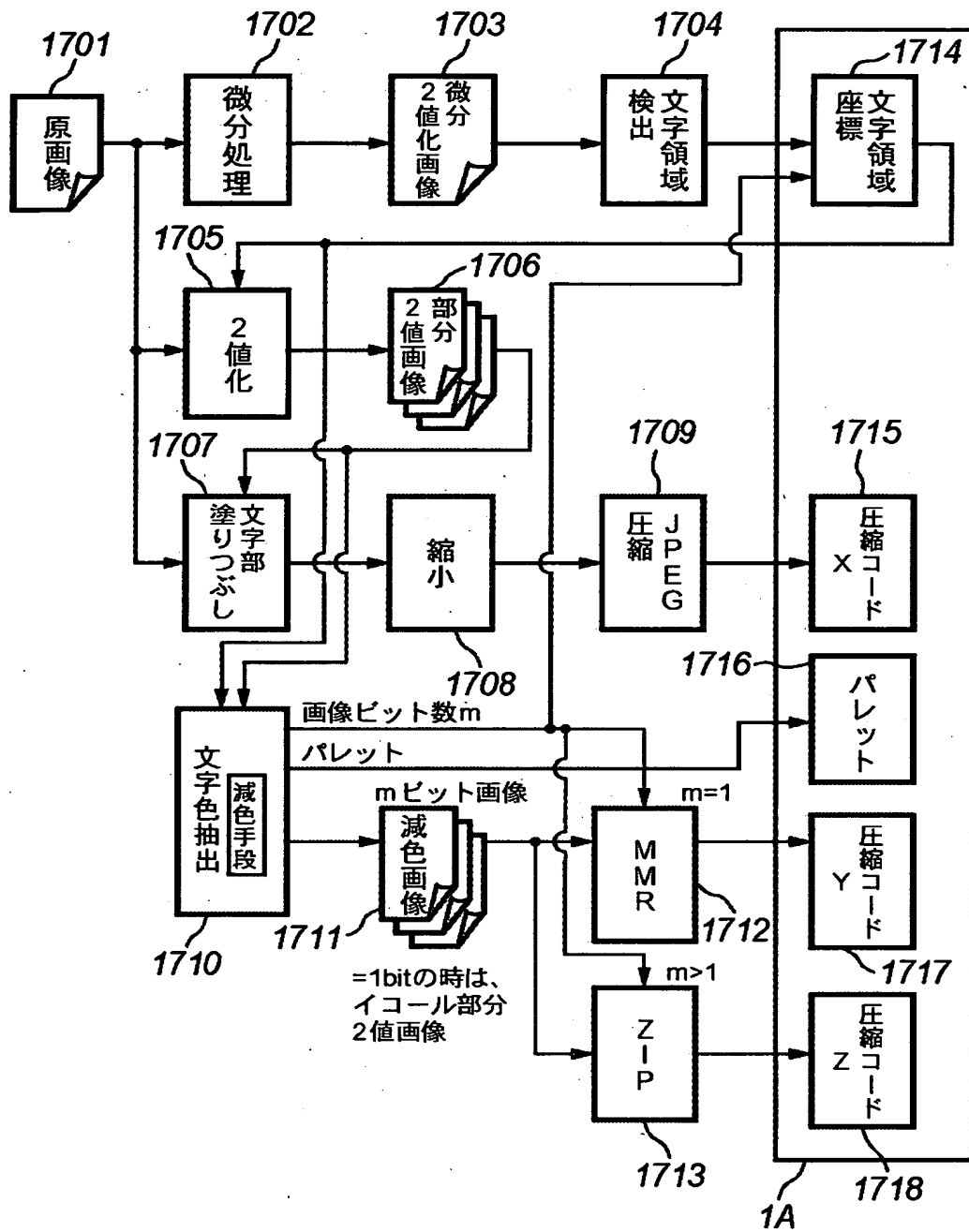
(20,30,225)の青

(c)

【図 1 5】



【図16】



【図 17】

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
		*		
+1	+1	+1	+1	+1
+1	+1	+1	+1	+1

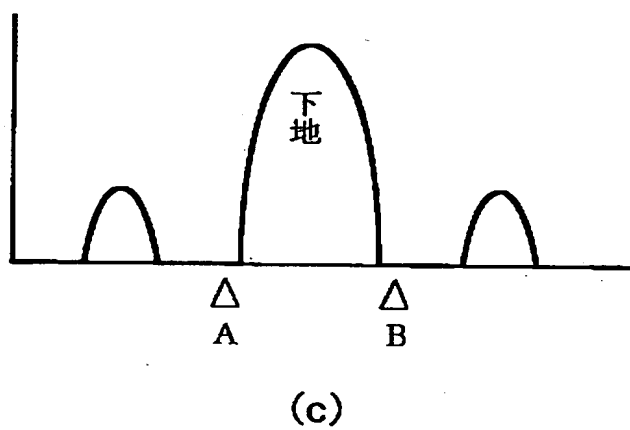
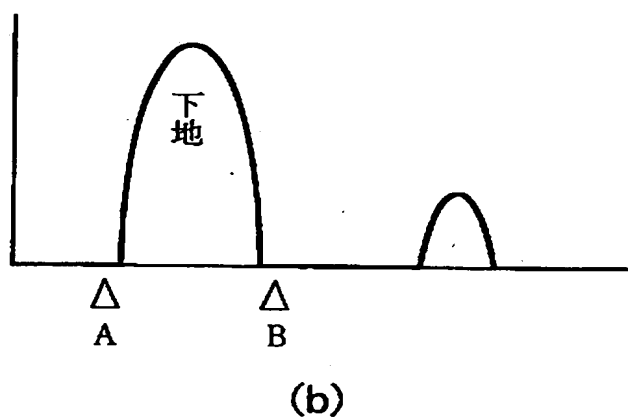
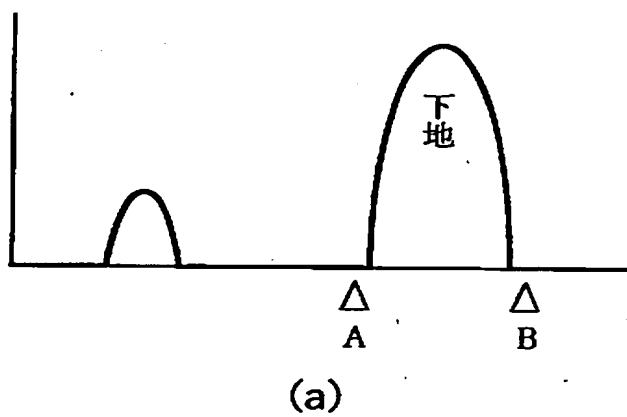
-1	-1		+1	+1
-1	-1		+1	+1
-1	-1	*	+1	+1
-1	-1		+1	+1
-1	-1		+1	+1

(a)

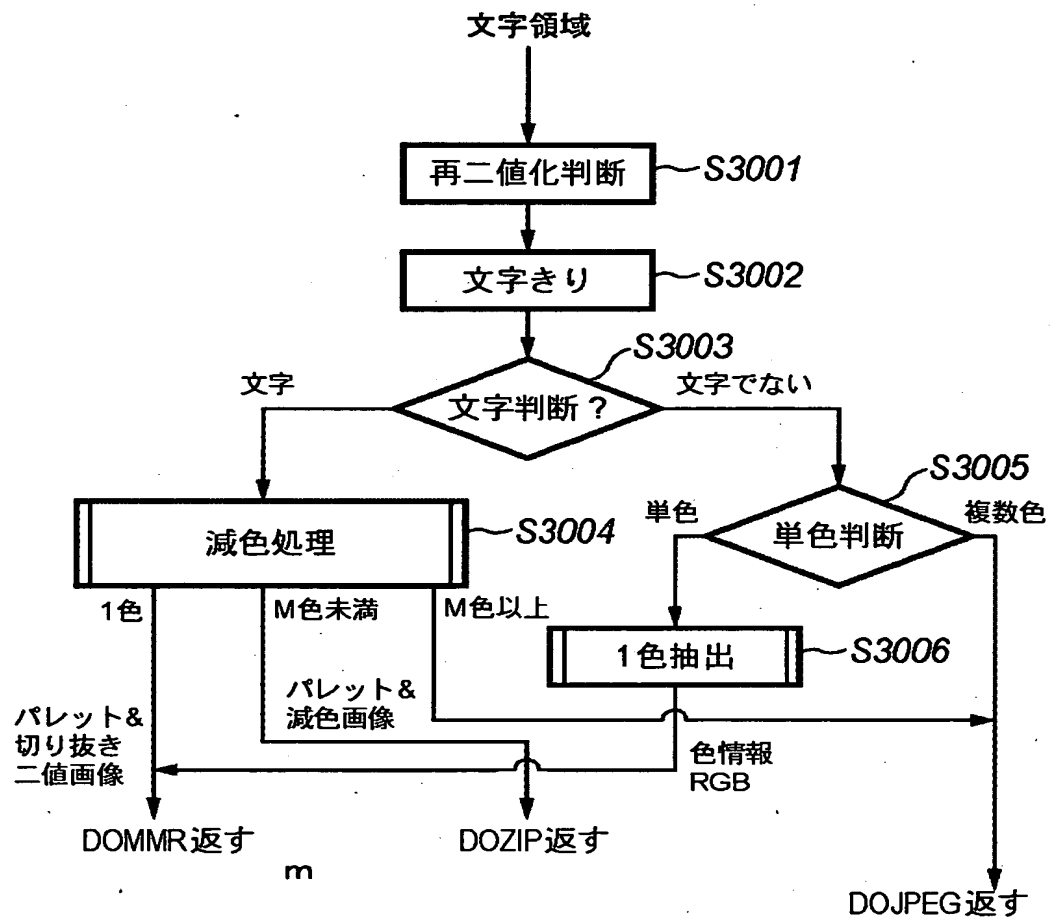
-1	-1	-1
-1	8	-1
-1	-1	-1

(b)

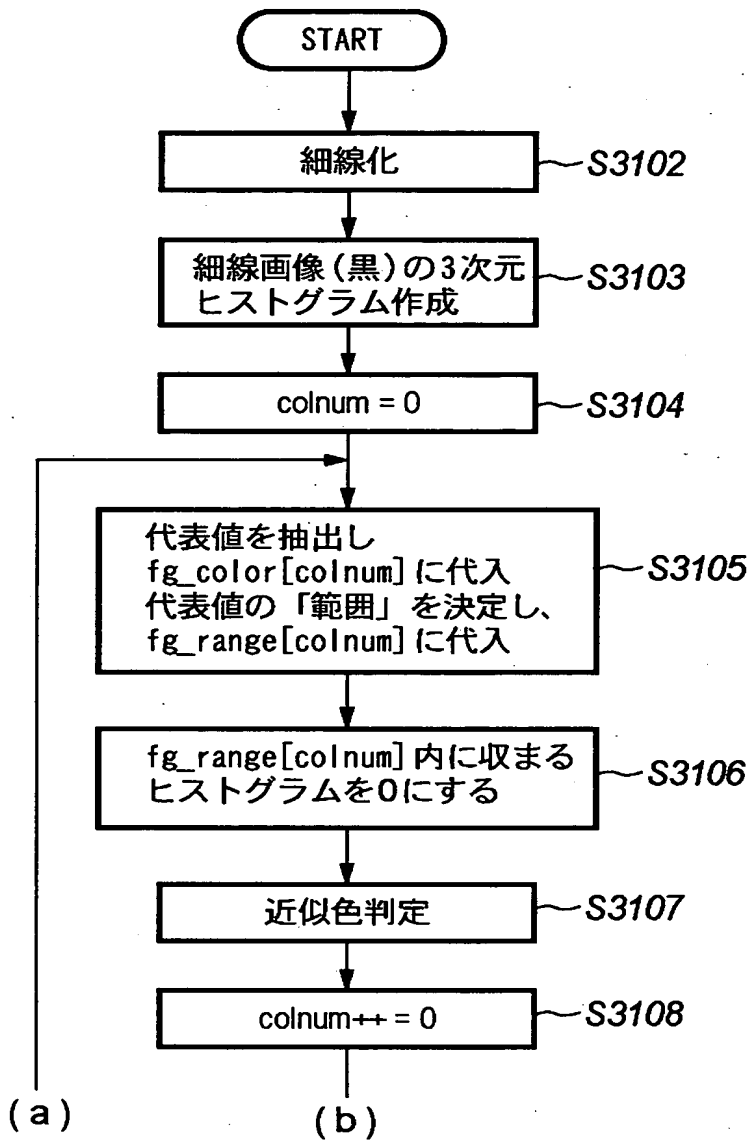
【図18】



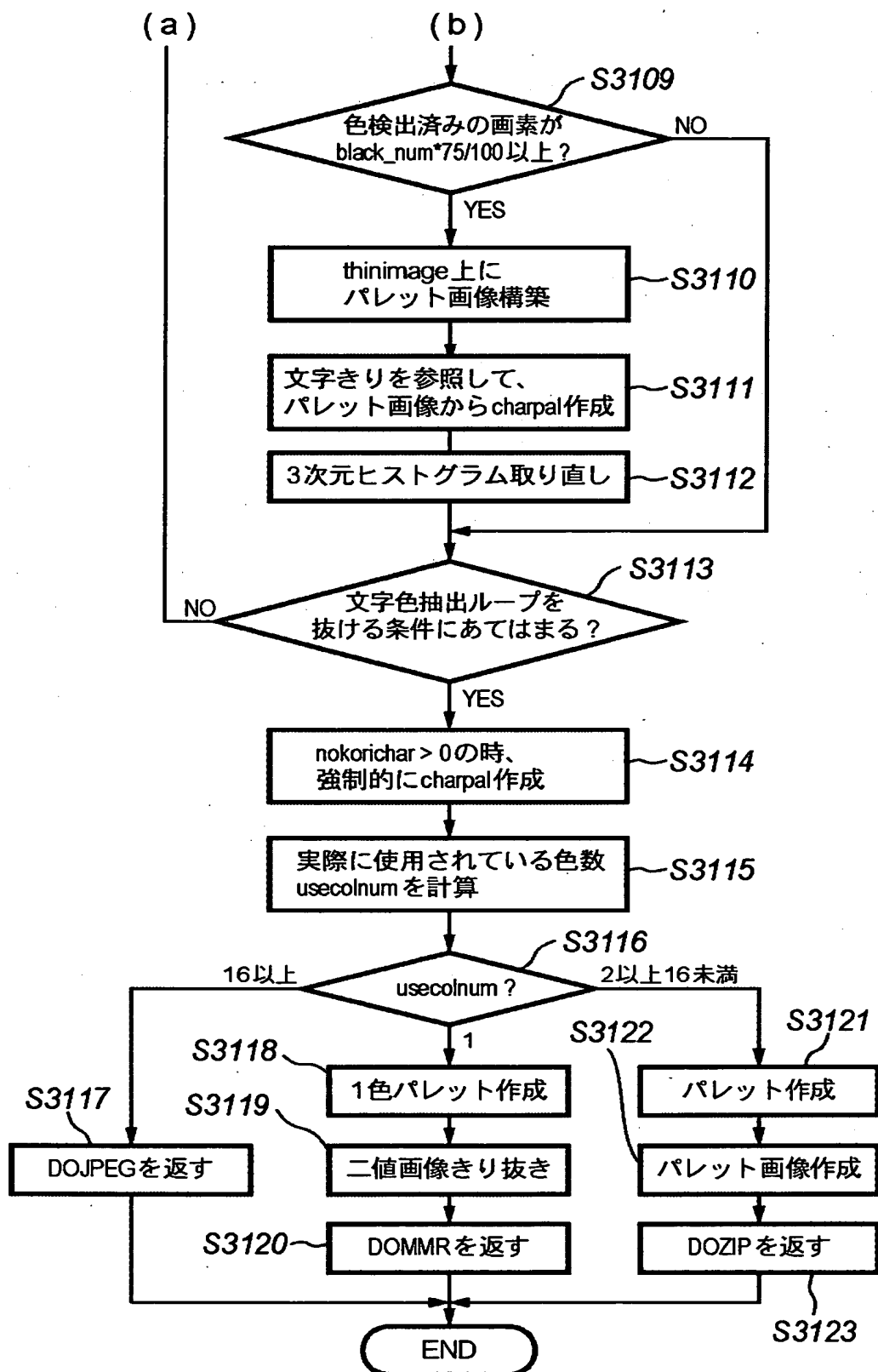
【図 19】



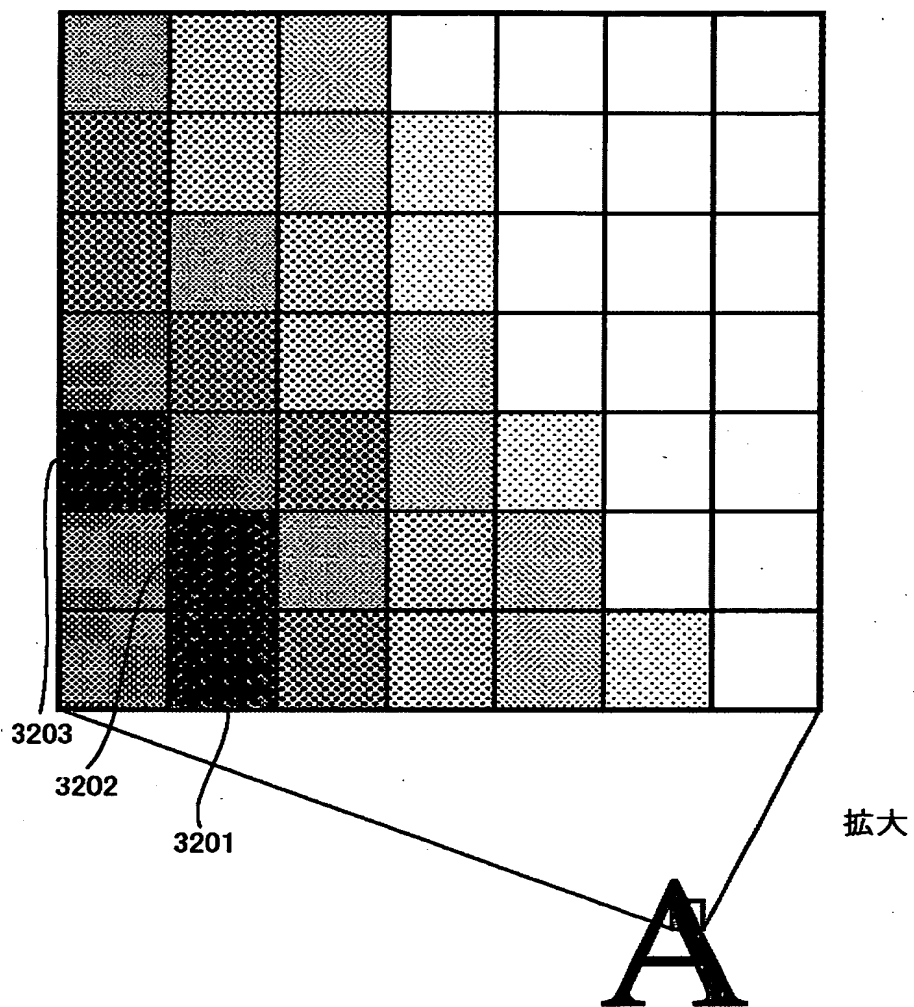
【図20】



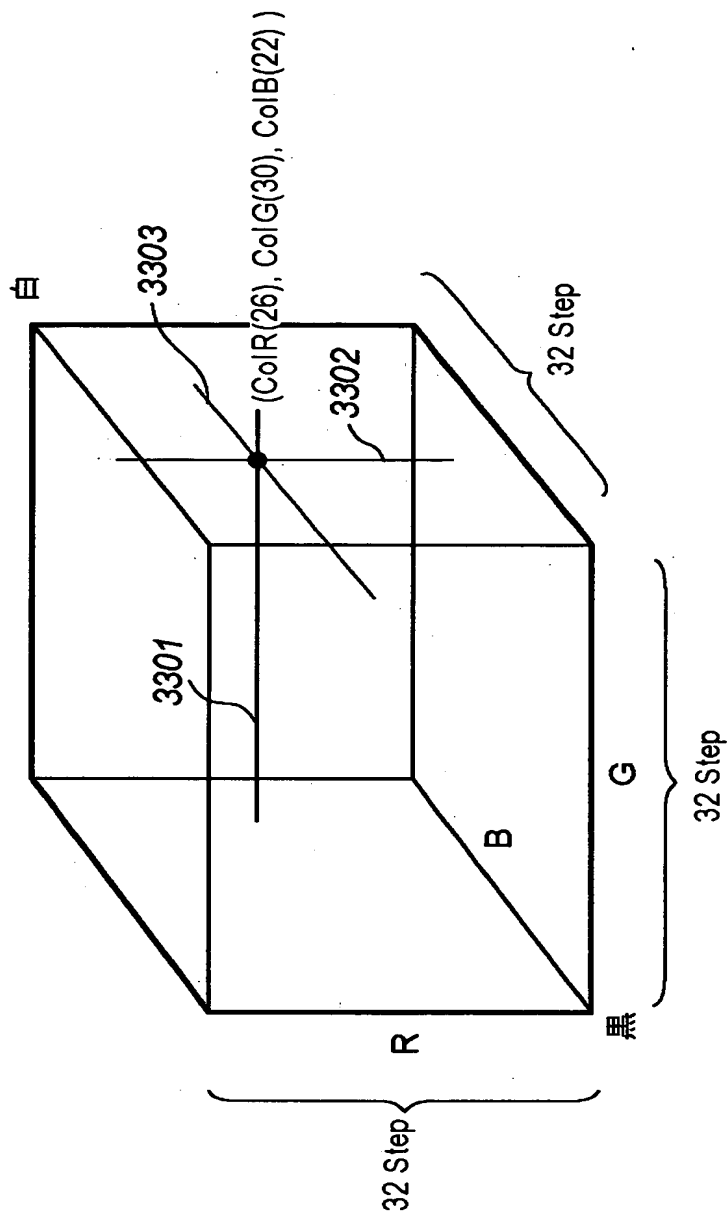
【図 2 1】



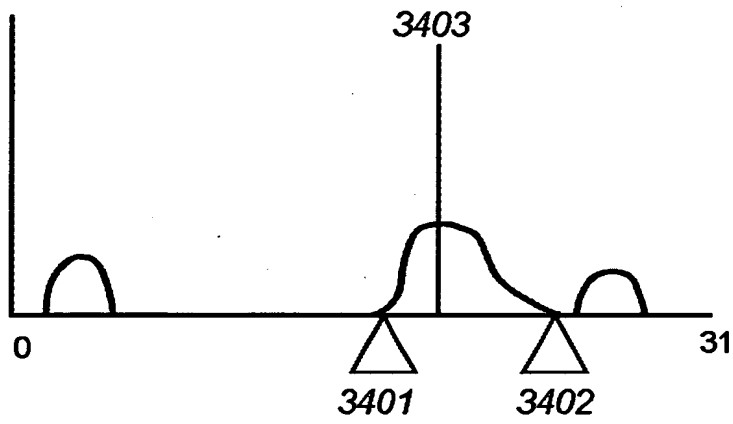
【図 2 2】



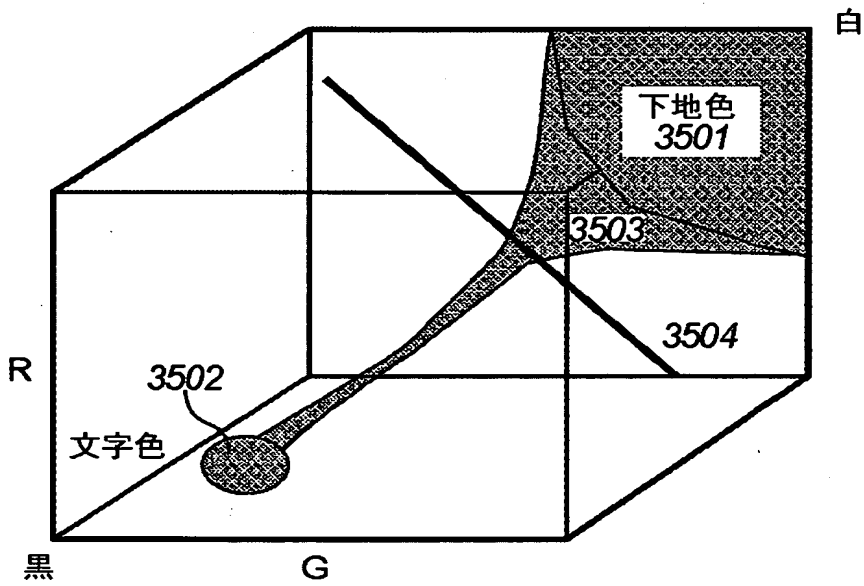
【図 2 3】



【図 24】

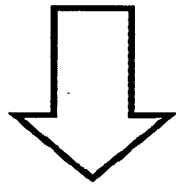


【図 25】



【図26】

明日は晴れます。
昨日は雨でした。



明日は晴れます。
昨日は雨でした。

【書類名】 要約書

【要約】

【課題】 文字領域に対し複数色を割当てて画像処理装置、画像処理方法並びにその記憶媒体を提供すること。

【解決手段】 カラー画像データから検出された文字領域に対し文字きり処理を行い（S 3 0 0 2）、文字領域の文字を構成する複数の色から、いくつかの代表的な色を導き出し、文字きり単位毎に代表的な色の内の 1 色を割り当てる（S 3 0 0 4）。

【選択図】 図 1 9

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日
[変更理由] 新規登録
住 所 東京都大田区下丸子3丁目30番2号
氏 名 キヤノン株式会社